1.1.1 Ölü Kod: İfadenin Daima False Olması (Dead Code: Expression is Always False) (CWE-570)

Açıklık Önem Derecesi: Düşük

Açıklığın Etkisi: Uygulama güvenliğinin sürdürebilirliğini azaltma

Açıklığın Barındıran Dosyalar/Satırlar:

Proje Dosyası/Dosya Adı	Satır Numarası

Açıklığın Açıklaması:

Uygulamalarda çalışmayan, işleve sahip olmayan kodlamalar yer alabilmektedir. Bu kodlar uygulamanın kod kalitesini düşürücü etkiye sahiptir. Uygulama zamanla büyüdükçe sonradan uygulama kaynak kodlarına bakıldığında her bir kodun işlevini anlama zorlaşır ve çalışmayan kodların varlığı da bu zorluğu arttırır. Bu durum uygulama güvenliğini sürdürme noktasında bir negatiflik olarak yansır. Uygulamalardaki çalışmayan kodlar uygulama güvenliğini sürdürme noktasında dezavantaj sunduklarından dolayı dolaylı yoldan güvenliğe dokunan bir açıklık olarak ele alınırlar.

Uygulama kaynak kodlarında ifadeler daima false olabilmektedir. Bu durum çalışmayan kod blokları meydana getirmektedir. Bunu örneklemek maksatlı Java dilinde iki ayrı örnek verilmiştir:

Java Güvensiz Kod Bloğu:

```
public void setUpCalls() {

    boolean firstCall = false;
    boolean secondCall = false;

    if (fCall > 0) {
        setUpFCall();
        firstCall = true;
    }
    if (sCall > 0) {
        setUpSCall();
        firstCall = true;
    }
    if (firstCall && secondCall) {
        setUpDualCall();
    }
}
```

Bu "ifadelerin daima false olması" açıklıklı örnekte if (firstCall && secondCall) bloğu daima false dönecektir ve içerisindeki setUpDualCall() metodu hiçbir koşulda çalışmayacaktır. Çünkü secondCall nesnesine en başta false değeri atanmaktadır ve blok boyunca da secondCall false olarak kalmaktadır. firstCall ise true da olsa false da olsa bloğun sonlarında yer alan if (firstCall && secondCall) ifadesinde secondCall her koşulda false kaldığından daima false olacaktır. Böylece if (firstCall && secondCall) bloğu hiçbir koşulda true olamayacaktır ve içine girilemeyecektir. Bu durum çalışmayan, fuzuli bir setUpForCall()'ı kodu meydana getirecektir.

Java Güvensiz Kod Bloğu:

```
public void setUpCalls() {
    boolean firstCall = false;
    boolean secondCall = false;

    if (fCall > 0) {
        setUpFCall();
        firstCall = false;
    }

    if (sCall > 0) {
        setUpSCall();
        secondCall = true;
    }

    if (firstCall && secondCall) {
        setUpForCall();
    }
}
```

Bu "ifadelerin daima false olması" açıklıklı diğer örnekte if (firstCall && secondCall) bloğu daima false dönecektir ve içerisindeki setUpDualCall() metodu hiçbir koşulda çalışmayacaktır. Çünkü firstCall nesnesine en başta false değeri atanmaktadır. Blok boyunca da firstCall false olarak kalmaktadır. İlk if bloğunda firstCall'a true atanacakken hataen false ataması tekrarlanmaktadır. Bloğun sonlarında yer alan if (firstCall && secondCall) ifadesi ise firstCall blok boyunca her koşulda false kaldığından daima false olacaktır. Böylece if (firstCall && secondCall) bloğuna hiçbir şekilde girilemeyecektir. Bu durum çalışmayan, fuzuli setUpForCall() kodunu meydana getirecektir.

Kurum uygulamada "ifadenin daima false olması (CWE-570)" açıklığı tespit edilmiştir:

::::::::BULGU:::::::::

Açıklığın Önlemi:

Uygulama kaynak kodlarında işlevsiz / çalışmayan kodlar giderilmeli. Örneğin bir karara bağlanıp çalışır hale getirilmeli veya kaldırılmalı.

Referanslar:

1. https://vulncat.fortify.com/en/detail?id=desc.structural.java.dead_code_expression_is_always_false#Java%2fJSP
2. https://cwe.mitre.org/data/definitions/570.html